Hindawi
Journal of Applied Mathematics
Volume 2023, Article ID 4573352, 13 pages
https://doi.org/10.1155/2023/4573352

Research Article

A Study of Deep Learning Neural Network Algorithms and Genetic Algorithms for FJSP

Xiaofeng Shang

School of Management, City Institute, Dalian University of Technology, Dalian 116600, China

Correspondence should be addressed to Xiaofeng Shang; shangxiaofeng@mjc-edu.cn

Received 28 October 2022; Revised 4 September 2023; Accepted 5 October 2023; Published 25 October 2023

Academic Editor: Waleed Adel

Flexible job-shop scheduling problem (FJSP) is a new research hotspot in the field of production scheduling. To solve the multiobjective FJSP problem, the production of flexible job shop can run normally and quickly. This research takes into account various characteristics of FJSP problems, such as the need to ensure the continuity and stability of processing, the existence of multiple objectives in the whole process, and the constant complexity of changes. It starts with deep learning neural networks and genetic algorithms. Long short-term memory (LSTM) and convolutional neural networks (CNN) are combined in deep learning neural networks. The new improved algorithm is based on the combination of deep learning neural networks LSTM and CNN with genetic algorithm (GA), namely, CNN-LSTM-GA algorithm. Simulation results showed that the accuracy of the CNN-LSTM-GA algorithm was between 85.2% and 95.3% in the test set. In the verification set, the minimum accuracy of the CNN-LSTM-GA algorithm was 84.6%, both of which were higher than the maximum accuracy of the other two algorithms. In the FJSP simulation experiment, the AUC value of the CNN-LSTM-GA algorithm was 0.92. After 40 iterations, the F1 value of the CNN-LSTM-GA algorithm remained above 0.8, which was significantly higher than the other two algorithms. CNN-LSTM-GA is superior to the other two algorithms in terms of prediction accuracy and overall performance of FJSP. It is more suitable for solving the discrete manufacturing job scheduling problem with FJSP characteristics. This study significantly raises the utilisation rate of the assembly shop's equipment, optimises the scheduling of FJSP, and fully utilises each processing device's versatile characteristics, which are quite useful for the production processes of domestic vehicle manufacturing companies.

1. Introduction

In the context of the intelligent era, all aspects of life are gradually moving towards intelligence, and the production mode of flexible work in industrial production is gradually becoming popular. Scheduling problem is one of the most critical problems in manufacturing process planning and management, and one of the most difficult problems in this area is the job-shop scheduling problem (JSP). Each workpiece is made up of a series of sequential restrictions that must be processed by a group of machines. Each process only needs a single machine, which is constantly accessible and capable of processing one operation at a time without pausing. To optimise a specific performance indicator, decisions must be made regarding the sequence in which the machine's processes should run. A typical performance metric for JSP is time to completion, the time it takes to get everything done. The flexible job-shop scheduling problem (FJSP) is an extension of the classic JSP and is more difficult than the traditional JSP. Each operation in this problem can be processed on any one of a set of available machines. In this context, the problem of flexible job-shop scheduling has become a problem of research significance, and there have been a lot of relevant researches [1]. Issues with scheduling are intrinsically tied to the growth of the manufacturing sector. Scheduling issues have arisen ever since the manufacturing sector was founded, even in the most sophisticated domains of contemporary manufacture. The main issues with production are schedule-related; without a scheduling system, industrial technology cannot fully express its sophisticated nature. Studying the production scheduling issue is more important, particularly for flexible assembly shops like the automotive industry with more complicated work flows. Current flexible operations are mainly single-objective flexible operations, and there is a relative lack of research on multiple-objective flexible operations [2]. In the current industrial pursuit of intelligence, the efficiency requirements of the flexible workplace are gradually increasing, and workers need to be able to work as efficiently as possible without being overly fatigued. In addition, there may be many uncontrollable factors such as emergencies in the flexible workshop. Therefore, the entire operation of the flexible workshop needs to be intelligently monitored [3]. Job-shop scheduling problem (JSP) is one of the most difficult problems in manufacturing process planning and management. In this problem, a group of machines is required to process a group of workpieces, each of which is formed by a sequence of sequential processes. Each operation requires only one machine and can be handled without interruption. Decisions include how to order processes on the machine to optimize given performance metrics. Versatile job-shop scheduling problem (FJSP), a variation of the standard JSP, is more challenging. This is so that FJSP can present the job path as a decision in addition to sorting. Which machine will handle the work in each process depends on the work path decision. However, the design of existing methods and algorithms requires large-scale feasibility test during the solution process, which directly leads to low efficiency and poor application performance of the algorithm. In light of these factors and the resulting need, the research of the FJSP problem has grown both in complexity and utility. Industrial production facilities are essential to industry, which is in turn the undisputed economic engine of a whole nation or region. Wang et al. proposed a multipopulation collaborative genetic algorithm based on collaborative optimization algorithm for FJSP problems. Good performance of the algorithm is guaranteed by adopting and designing good operators [4]. An artificial neural network was developed to predict the flow of single-walled carbon nanotube (CNT) nanofluids towards a boundary layer of parabolic, conical, and cylindrical nonlinear isothermal needles with convective boundary conditions. The results showed that the artificial neural network could predict the Nusselt number and the surface friction coefficient with high accuracy [5]. Shafiq et al. used the numerical analysis results obtained in four different scenarios to develop multilayer artificial neural networks. The study showed that neural networks are a powerful and effective mathematical tool, which could be used for reliability analysis of hybrid models [6]. The research combines deep learning neural network algorithms with genetic algorithms. And a new and improved algorithmic model is constructed for monitoring changes and making certain expectations and judgements throughout the flexible job shop. At present, heuristic algorithm has become the main means to solve job-shop scheduling problem, and genetic algorithm is the most used. However, genetic algorithm has some problems, such as large randomness and low searching efficiency. In order to solve this problem, deep learning neural network is proposed to improve the genetic algorithm, improve the convergence speed and search efficiency of the population, and then realize the solution of the dynamic scheduling optimization model. The goal of this study is to maximise flexible assembly workshop scheduling, increase equipment utilisation rates, and fully utilise the flexible benefits of each processing equipment. Consequently, shortening the manufacturing cycle of assembled items will aid businesses in being more efficient and competitive.

2. Related Work

Many domestic and foreign scholars had made research on LSTM neural networks in deep learning as well as genetic algorithms. Traditional transient stability evaluation methods were time-consuming. To solve the problem, Shahzad proposed two different methods, namely, artificial neural network (ANN) and support vector machine (SVM), to evaluate transient stability. The results showed that ANN training precision was higher than SVM, and the training time was less than SVM. Therefore, the classification and computing performance of ANN were superior [7]. Fard compared multiple artificial neural network (ANN) methods to better predict the number of COVID-19 cases at N steps ahead of the current time. Research results showed that in most cases, long short-term memory (LSTM) had the highest prediction accuracy for COVID-19 cases [8]. Szarek proposed a novel hybrid convolutional neural network model that uses a 1-to-1 method to predict the occurrence of four types of solar flares within 24 hours. The results showed that the prediction effect of this model was better than that of the traditional model [9]. Chea and Nam propose an optimal residual deep neural network and an efficient image preprocessing technique and apply the method to three eye disease classifications in currently available public datasets. The study showed that the method had high peak and average accuracy and can be used as an assistive device in future computer-assisted clinical applications [10]. Zheng et al. proposed a hybrid deep convolutional neural network model and applied it to solar flare prediction. The results illustrated some of the breakthrough key features that could be automatically extracted by the model and might provide important clues for studying the mechanisms of flares [11]. Bhola et al. introduced optimal genetic algorithms in their study of wireless sensor network optimization to find the optimal path using its fitness function [12]. Xue proposed a node selection method based on genetic algorithm for the optimal selection of wireless sensor nodes in the Internet of Things. The results showed that this method can guarantee the coverage of the monitored area, reduce the network energy consumption, and maintain the balance of network energy consumption [13]. To solve the problem of low classification accuracy in evaluating chronic kidney disease problems, Sahu proposed a feature selection technique based on genetic algorithm. The results showed that the technique improved the classification accuracy and could be used for the early identification of chronic kidney disease (CKD) [14].

The FJSP problem has also been studied by researchers. In order to ensure green production while maximising production efficiency, Gong et al. proposed a new nondominated integrated adaptive ranking algorithm for the multiobjective flexible job-shop scheduling problem.

Table 1: Various algorithms and FJSP problem solving.

Year	Author	Method	Advantage
2021	Wang et al. [4]	Multipopulation collaborative genetic algorithm based on collaborative optimization algorithm for solving FJSP problems	The algorithm has good recommendation performance
	Szarek [9] and Zheng et al. [11]	A new hybrid convolutional neural network model	The predictive performance of this model is superior to traditional models
	Chea and Nam [10]	Optimal residual depth neural network image processing technology	This method has high peak and average accuracy
	Bhola et al. [12]	Introducing optimal genetic algorithm in the optimization research of wireless sensor networks	Finding the optimal path through its fitness function
	Sahu [14]	Feature selection technology based on genetic algorithm	This technology improves classification accuracy
	Gong et al. [15]	A new nondominated synthetic adaptive sorting algorithm	Optimizable multiobjective flexible job shop scheduling
	Tan et al. [16]	A flexible job-shop scheduling scheme with dual resource constraints	This method enhances local search functionality and achieves better scheduling
	Wang et al. [17]	A hybrid algorithm based on grey wolf and invasive weeds	This algorithm can effectively solve the flexible job-shop scheduling problem
2022	Shafiq et al. [6], Shahzad [7], and Fard [8]	Prediction method based on artificial neural network	This method has high prediction accuracy

Simulation results showed that the algorithm had better performance compared to other multiobjective algorithms [15]. To minimise maximum worker fatigue and maximum completion time, Tan et al. proposed a dual-resource constrained flexible job-shop scheduling scheme capable of identifying worker fatigue levels by constructing a multiobjective optimisation model. It also enhanced the local search capability. Experimental results showed that workers were better scheduled in the constrained flexible job shop [16]. To increase productivity in the flexible store, Wang et al. used the grey wolf algorithm and the green flower algorithm. With the addition of an efficient initial population generating mechanism, this strategy could enhance both local and global search capabilities as well as the calibre of the first answer. The outcomes demonstrated that, within a limited range, the algorithms could successfully solve the FJSP issue [17]. From the above studies, it can be seen that deep learning neural networks had been applied in various fields and had good compatibility with the FJSP problem, although there has not been a lot of research on the FJSP based on deep learning neural networks and genetic algorithms. In order to create new and improved algorithms that might handle the multiobjective FJSP issue, this work integrated genetic algorithms with LSTM and CNN. As shown in Table 1, the advantages of various algorithms and the solution of FJSP problems are shown.

3. Application of Genetic Algorithm and Deep Learning Algorithm in Flexible Job-Shop Scheduling Problem

3.1. Generation and Operation of Genetic Algorithm in Flexible Job-Shop Scheduling Problem. Flexible job shop manufacturing includes the flexibility of machines, processes, workflows, product quantities, and product types, and the related scheduling issues revolve around these factors. For FJSP, the two most important issues are machine selection and process sequencing. FJSP can be divided into two categories, namely, fully flexible job-shop scheduling and biased flexible job-shop scheduling. Fully flexible job-shop scheduling means that any machine can be selected for the entire production and processing of the product [18, 19]. Biassed flexible job-shop scheduling allows for the selection of machines for some procedures. But for the remaining steps, only particular machinery can be employed. Because flexible job-shop scheduling is better in line with actual production needs, it is the foundation of the FJSP. The operation symbols and abbreviations are shown in Table 2.

The FJSP has four properties: multiconstraint, discrete, multiobjective, and computational complexity [20]. It is important to ensure that the problem is ordered, taking into account multiple factors such as emergency situations and, in particular, the need to consider multiple variations. The study, therefore, begins with a genetic algorithm (GA), which is inherently suited to the study of highly variable objectives and can also be applied to multiobjective problems. A GA is a bionomic algorithm derived from evolutionary theory in biology that finds optimal solutions by mimicking the evolution of organisms in nature [21]. Coding chromosomes, fitness assessment, initial population setting, selection, variation, crossover, and termination principle are the seven essential components of the GA that make up the complete process. The basic flow of the algorithm is shown in Figure 1.

As shown in Figure 1, the main role of the encoding chromosome is to encode the data parameters that need to be mined for computation. The encoding chromosome will

Table 2: Operation symbols and abbreviations of the research algorithm.

Operator symbol				Abbreviation	
f	The objective function	v	The scale parameter	LSTM	Long short-term memory
ω	The weight	θ	The mapping of the Sigmoid function	CNN	Convolutional neural network
λ	An adjustable parameter	t	Sigmoid mapping of the function	GA	Genetic algorithm
p_i	The probability of selection of the ith individual	σ	The sigmoid function	FJSP	Flexible job-shop scheduling problem
$f(a_i)$	The individual fitness	W_f	The learnable weight parameter	ANN	Artificial neural network
x_a, x_b	Generation population	b_f	The bias vector parameter	SVM	Support vector machine
η	The core of the Laplace distribution	W_o	The weight of the output gate	ROC	Receiver operating characteristic
μ	The location parameter	b_o	The bias of the output gate	F1	$F1$ score

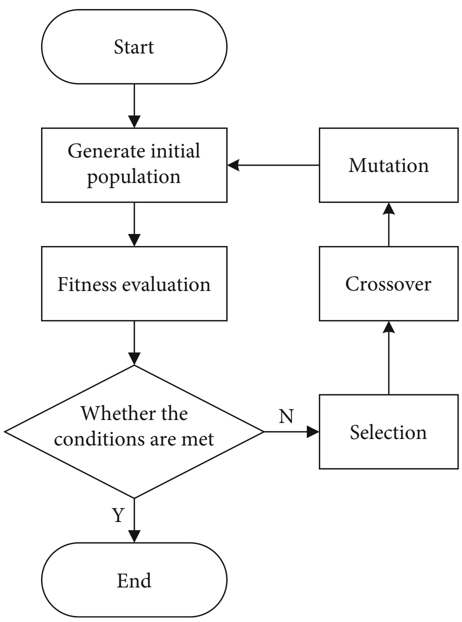


Figure 1: Basic flow chart of genetic algorithm.

symbolize the problem and define it in continuous space using strings. The initial population setting uses an algorithm to define the initial population and select a suitable initial population, the number of which must be appropriate. On the other hand, fitness evaluation uses the objective function to evaluate the outcome of the solution and compares the solution value to the value of fitness, which is the main basis for the selection mechanism. Selection, crossover, and variation are all algorithmic steps that make use of the imitation of changes in genes in biology. Selection focuses on judging the goodness of an individual, i.e., comparing individual fitness values with the whole population and filtering out the relatively partially optimal solution. Crossover is a random exchange of information about the encoded chromosomes, and new individuals are constantly generated to enhance the search capability of the algorithm. The role of crossover in the algorithm is crucial. And the search power of the whole algorithm changes significantly if crossover is repeated to the end. Mutation is a random change in the chromosomal information of the population to form new chromosomes with a certain probability. By maintaining the global diversity of the population, variation ensures global search power. On the other hand, the termination principle outputs and terminates the algorithm after the results it produces meet certain principle requirements [22]. The simple process of variation is shown in Figure 2.

Figure 2 shows that during development, certain mutations can occur on target chromosomes, creating mutant chromosomes. After mating and selection, the gene values are changed with probability to form a new individual at some loci of a parent individual gene chain. In the genetic algorithm solution, a suitable fitness function must be selected and solved. According to the actual production needs, a multiobjective fitness function with weighting is designed as shown in equation (1) [23].

$$\text{fit} = \frac{1}{\omega_1\lambda_1 f_1 \cdot \omega_2\lambda_2 f_2 \cdot \omega_3\lambda_3 f_3 \cdot \omega_4\lambda_4 f_4}. \tag{1}$$

In equation (1), f represents the objective function, the number of multiobjective functions is set to 4, ω is the weight and the sum of the four weights is 1, and λ is an adjustable parameter. To enhance the intuition of the initial population size, the study adopts the natural number coding method for chromosome coding. The initial population size is set to 100, and the maximum number of iterations is set to 200. After the population has entered the selection operator, the chromosomes with the top 30% fitness are selected in the final generation using the ranked selection method. And the remaining individuals are selected using the spinning wheel method. The probability of the remaining individuals being selected is calculated as shown in the following equation:

$$P_i = \frac{f(a_i)}{\sum_{i-1}^{K} f(a_i)}. \tag{2}$$

In equation (2), p_i represents the probability of selection of the ith individual, K represents the population size, and $f(a_i)$ represents the individual fitness. In the crossover operation, the parental chromosomes are crossed over and swapped. According to the characteristics of FJSP, the

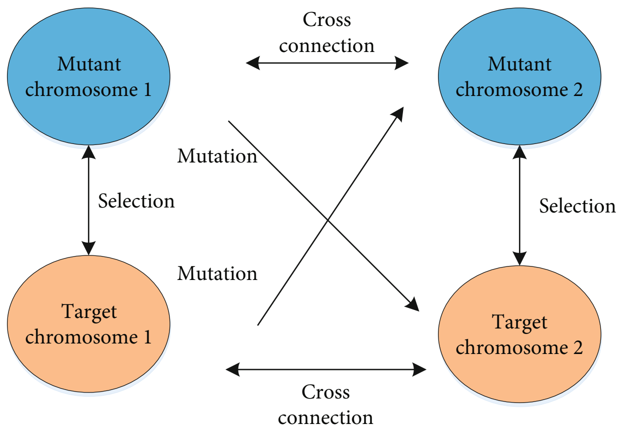


Figure 2: Simple map of coding chromosome mutation.

study uses the Laplace crossover operator to perform crossover operation on x_a and x_b of a certain generation. And its obtained next generation is shown in the following equation:

$$\begin{cases} x_a' = x_a + \eta|x_a - x_b|, \\ x_b' = x_b + \eta|x_a - x_b|. \end{cases} \tag{3}$$

In equation (3), η is the core of the Laplace distribution, and the core parameters are calculated as shown in equation (4). In equation (4), μ is the location parameter, and v is the scale parameter, both of which can be obtained backwards by quotienting the distribution of functions [24].

$$\eta = \begin{cases} \mu - v \ln(u), u \geq \dfrac{1}{2}, \\ \mu + v \ln(u), u < \dfrac{1}{2}. \end{cases} \tag{4}$$

The position parameter is mainly used to determine the distance to the individuals of the parent population, while the scale parameter is used to express the value of the distance to the parent. Under the combined effect of the two parameters, the distance between individuals of the parent affects the distance of its offspring, making the offspring a proportional extension of the parent. The GA can, therefore, extrapolate the parents from the children in the FJSP to obtain accurate information about the parents.

Encoding and decoding refer to the conversion between chromosomes and scheduling solutions, which is the first and key problem in the successful implementation of genetic algorithm optimization. For traditional job-shop scheduling problems, most studies use process-based coding. However, the issue of flexible job-shop scheduling also requires choosing the right machine for each working procedure in addition to figuring out the machining order of the working method. Therefore, for flexible job-shop scheduling problem, the coding of genetic algorithm consists of two parts. The first part is process-based coding, which is used to determine the processing sequence of the process. The second part is the coding based on machine assignment, which is used to select the machining machine of each I-sequence. By combining these two coding methods, it can obtain a feasible solution to the flexible interjob scheduling problem. Chromosomes combining the two coding modes are shown in Figure 3.

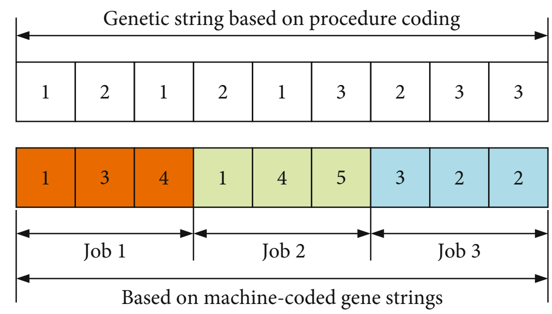


Figure 3: Chromosomes combining the two coding modes.

Figure 3 shows that the sequence of I-sequence processing is determined by the gene string based on process coding, and the processing machine for all processes of each workpiece is determined by the gene string based on machine coding. Unlike the encoding process, decoding is divided into general decoding or semiactive decoding, active decoding, and fully active decoding. Therefore, the semiactive decoding method is used for chromosome decoding.

3.2. The Improvement of Deep Learning LSTM and CNN in Flexible Job-Shop Scheduling Problem. In order to achieve more accuracy on the results of FJSP operations and to achieve a balance between local and global, the study simultaneously uses deep neural network algorithms commonly used in artificial intelligence. In order to achieve superior outcomes on the FJSP problem, this study integrates the two approaches. Artificial neural network algorithms, which include intricate mechanics, are the source of deep learning. Multiple neural networks can be used by deep learning neural networks as their own subnetworks, and hidden layers can be constructed from these subnetworks. The number of implicit layers is large, and each implicit layer can be linearly transformed to the output of the higher layer network. As a result, deep learning neural networks are able to solve complex problems with more inputs and more accurate representations with a larger set of functions. And complex problems such as FJSP are a good fit for deep learning neural networks [25].

The frameworks for deep learning neural networks include deep confidence networks, convolutional neural networks, deep neural networks, and recurrent neural networks. The FJSP problem requires a high degree of mastery of the specific time of change and a certain degree of image monitoring capability. In view of this, the deep learning neural network studied uses both LSTM and CNN. The structure of the LSTM neural network is based on the classical recurrent neural network, introducing three logical structures of input gate, output gate, and forgetting gate. And its basic structure is shown in Figure 4.

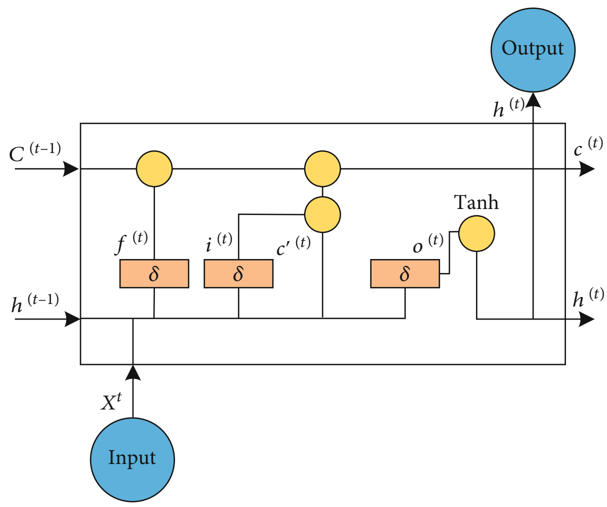


Figure 4: Schematic diagram of LSTM neural network structure.

In Figure 4, in the LSTM network structure, input gates, output gates, and forgetting gates are denoted by i, o, and f, respectively. C is the memory cell, the input data is X, and the implicit state is H. The input contains the cell state at a specific time step and the previous implicit level $h^{(t-1)}$. The control function determines whether the information is cleared or retained. The final output cell state vector $f^{(t)}$ takes the value interval [0,1]. If the value is 1, then the input value is retained as a whole. If the value is 0, then the value is removed as a whole [26]. The input determines whether the information is added to the cell state for the data update. The sigmoid function is used to remove the information of a particular time step and the previous hidden layer. The current input cell state is then resolved, and a postselection vector with values in the range [-1, 1] is created after the tach function. Finally, the current moment's cell state $c^{(t)}$ is calculated, which is first multiplied by the previous moment's cell state $c^{(t-1)}$ and the oblivion gate and then multiplied by the input gate $c'(t)i^{(t)}$ to obtain the final result. The output gate is responsible for selecting the valuable cell state that will be displayed for output, and its particular implementation procedure entails two parts. A filter is first acquired. The cell state vector's value is then compressed to the range [-1, 1] using the tach function, while the result obtained by multiplying the vector and the filter is used as the basis for determining the hidden information. To train the algorithm, the last layer's error is first calculated. A gradient descent algorithm is then used to update the settings. Once all parameters have been modified, they are then transmitted along one at a time. There are eight sets of parameters in the long and short-term memory network that must be learned: the forgetting gate, the input gate, the output gate, the weight matrix of the unit states, and the bias term. They are calculated differently in the two directions of back propagation [27]. Sigmoid function and the tach function are calculated as shown in equations (5) and (6), respectively [28, 29].

$$\sigma_\theta(u) = \frac{1}{1 + e^{-\theta^u}}, \tag{5}$$

$$g(u) = \frac{1}{1 + e^{-u}}. \tag{6}$$

In equation (5), θ is the mapping of the sigmoid function. The expression of the forgetting gate at t is shown in equation (7). In equation (7), σ is the sigmoid function, W_f is the learnable weight parameter, and b_f is the bias vector parameter.

$$f_t = \sigma\left(X_t W_{\mathrm{xf}} + H_{t-1} W_{\mathrm{hf}} + b_f\right). \tag{7}$$

The information to be stored in the neural unit is determined, and the value of the update is initially determined by the sigmoid function network layer. Its expression is calculated as shown in equation (8). In equation (8), W_i is the weight of the update gate, and b_i is the bias of the update gate.

$$i_t = \sigma(X_t W_{\mathrm{xi}} + H_{t-1} W_{\mathrm{hi}} + b_i). \tag{8}$$

Using the tanh function, candidate values are generated through the hyperbolic tangent function tanh layer. The expression is calculated as shown in the following equation:

$$\bar{C}_t = \tanh\,(X_t W_{\mathrm{xc}} + H_{t-1} W_{\mathrm{hc}} + b_c). \tag{9}$$

The memory state is then updated. The state is updated by dot product operations, using output and forgetting gates to control the flow of information. Finally, the updated state is obtained, as shown in the following equation:

$$C_t = f_t \Theta \tanh\,(C_t). \tag{10}$$

When the forgetting gate and 1 are close and the input gate is close to 0, the memory unit of the old state is stored at the current moment. At this point, the LSTM network can handle the disappearance of the gradient in the recurrent neural network and reduce the error. Finally, the memory unit of the output state of the output gate is determined by the sigmoid function as shown in the following equation:

$$O_t = \sigma(X_t W_{\mathrm{xo}} + H_{t-1} W_{\mathrm{ho}} + b_o). \tag{11}$$

In equation (11), W_o represents the weight of the output gate, and b_o represents the bias of the output gate. t is the time-implicit layer state, and H_t is calculated as shown in the following equation:

$$H_t = O_t \Theta \tanh\,(C_t). \tag{12}$$

In contrast, the structure of a CNN is divided into an input layer, a convolutional layer, an activation function, a pooling layer, and a fully connected layer [30, 31]. The

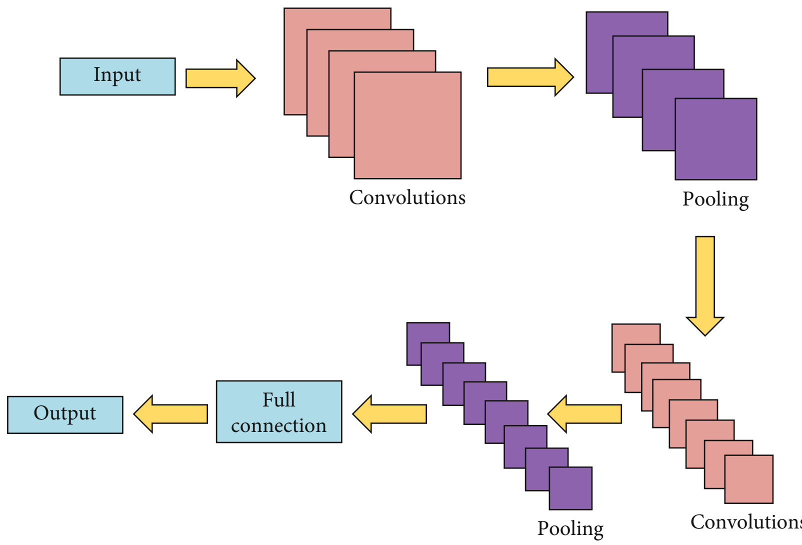


Figure 5: Basic structure diagram of CNN.

input layer is a pixel matrix that performs various operations on the sample data, including data normalization, dimensionality reduction, pixel correction, and scale normalization. The convolution layer contains multiple feature data, which are processed locally by learning feature expressions using local sensing for each corresponding feature data. Each local is then subjected to a synthesis operation that combines the information from each local and obtains the global information through a convolutional operation. The activation layer is operated by an activation function, which generally uses the sigmoid function and the tach function. The method of activation function operation is nonlinear mapping, when the convolutional layer can extract more abstract features and thus enhance the function of the convolutional neural network. The structure of the entire CNN is shown in Figure 5.

In Figure 5, the basic structure of CNN includes input layer, convolution layer, pooling layer, fully connected layer, and output layer. Among them, the convolution layer and the pooling layer together form the hidden layer. The pooling layer influences the parameters of the fully connected layer. The fully connected layer is usually at the very back of the whole CNN structure and usually has several layers. The main function of the fully connected layer is to transform each local feature extracted by the convolutional layer into a whole by means of a weighting operation, thus obtaining a more complete and hierarchical overall feature. If the initial feature map of each of the input convolutional layers is x_j, the convolutional operation is shown in equation (13). In equation (13), $f(x)$ is the activation function, M_j is the set of initial feature maps, and i is the matching result. k_{ij} is the convolution kernel for the input of the ith initial feature map and the output of the jth initial feature map.

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1} \cdot k_{ij}^l + c_j\right). \tag{13}$$

The sensitivity at each node is first found, and then its size is calculated θ. And the sensitivity size is used to derive the corresponding parameters required for the l layer. The sum of the sensitivity values defined by interest from the connectivity layer l to $l+1$ is θ_j^{l+1}, which is multiplied by the corresponding weights W. Then, the activation function is obtained by taking the inverse of $f(u^l)$, as shown in equation (14). In equation (14), u is the input value of the neurons in layer l.

$$\theta_j^l = \theta_j^{l+1} W_j^{l+1} \cdot f\left(u^l\right). \tag{14}$$

The CNN-LSTM-GA algorithm, created by combining a deep learning neural network and a genetic algorithm, is then tuned using an adaptation function. The entire algorithm is shown in Figure 6.

The FJSP optimization process based on the improved genetic algorithm is shown in Figure 6. Firstly, parameters are set, such as the total number of genes in an individual, the total number of genes in a single individual, and the rescheduling cycle. Initialization is an experience pool used to train neural networks and initialize the evaluation of neural network models. The number of initial learning schedules is 1, and the number of learning steps is 1. At the initial scheduling time, the arrived workpiece is scheduled, and the current initial system state is recorded. Then, decode and measure the objective function value of the obtained population to generate the initial population. Decode and measure again, and if the number of iterations reaches the maximum, the deep learning neural network is initialized. Otherwise, the initial population is generated again. The deep learning neural network is trained. The anticipated value is output, and the procedure is completed if the training fulfils the expectation. Otherwise, keep using the deep learning neural network model to train. The projected value is output if the timetable is followed exactly. If not, reschedule. Repeat the process after initialising the neural network

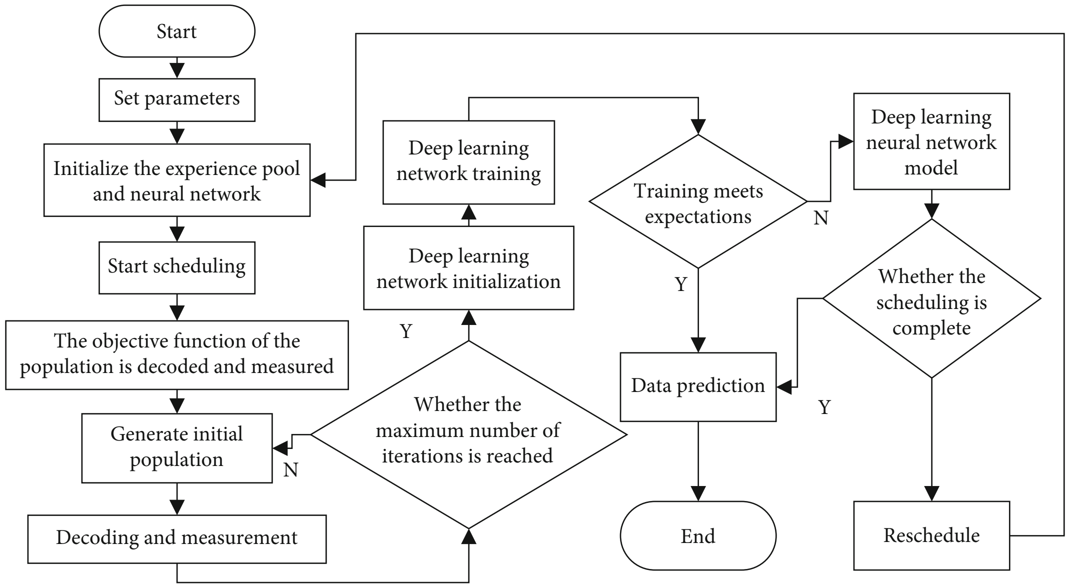


Figure 6: Improved algorithm flow chart.

Table 3: Experimental parameters and experimental environment settings.

Experimental environment		Experimental parameter	
Processor	Intel Core i5-520OU	Population size	100
Dominant frequency	2.2 GHz	Maximum iterations	100
Hard disk	2 T	Reward function	1
Internal memory	16 G	Learning rate	0.75
Programming environment	Python	Learn discount rates	0.2
Language	MATLAB	Global machine selection probability	0.6

and experience pool. The study uses 70% of the sample data as the training set, 15% of the remaining 30% of the data as the test set, and 15% as the sample set to obtain more informative results. To compare the performance of the CNN-LSTM-GA algorithm, the study will also compare this algorithm with two other algorithms, namely, CNN-LSTM and LSTM-GA. At the same time, these algorithms are compared with the same training method and the same sample data used. The sample data needs to be preprocessed first, as shown in equation (15). In equation (15), $\widehat{a}^p$ is the normalized data feature value, a^p is the original feature, $\bar{a}^p$ is its mean, and S^p is its variance.

$$\widehat{a}^p = \frac{(a^p - \bar{a}^p)}{S^p}. \tag{15}$$

4. Experimental Results and Analysis under the Comparison of Multiple Algorithms

The genetic optimization algorithm based on deep learning neural network algorithm is implemented on MATLAB 2018a. In order to verify the performance of the design model in this study, the scheduling data of a factory was collected as the data set of this test. The dataset has 1000 sample data sets, of which 60% is used as the training set and 40% is used as the test set. Firstly, the training set data is input into each algorithm network for training. The target error of training is set to 0.001, and the learning rate is 0.01. The optimizer uses Adam. Finally, the training errors of the three networks are compared to examine the performance progress of the proposed algorithm. Its specific experimental environment and experimental parameter settings are shown in Table 3.

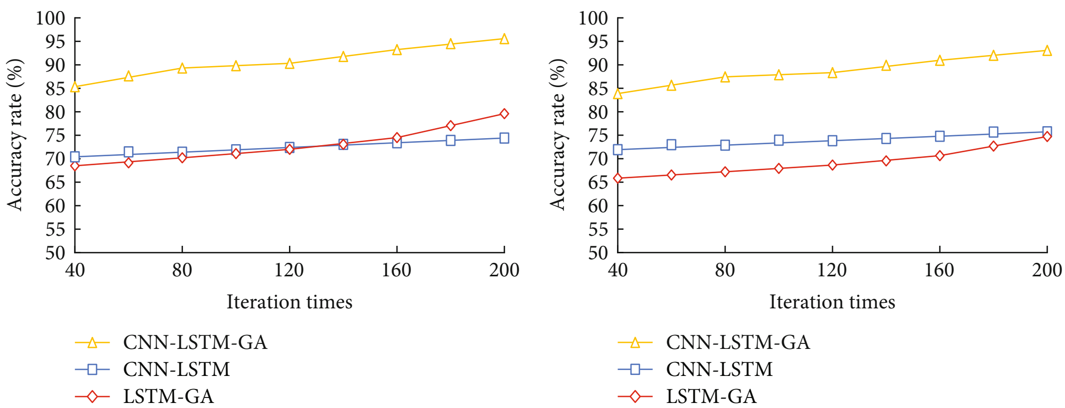
The research algorithm is compared with two other advanced algorithms in the case scale of 4×5, 10×10, and 10×15, respectively. AFI, ASI, and AFS represent the average robustness, average stability, and overall improvement of the algorithm, respectively. A negative value indicates improvement, while a positive value indicates deterioration. The results of the performance comparison of the four algorithms are shown in Table 4.

In Table 4, compared with the other two algorithms, the research method has a better improvement effect on the problem of different scale instances. The AFI, ASI, and AFS are lower than the other two algorithms, indicating the least deterioration degree. And the algorithm performance tests focus on accuracy, recall, precision, ROC and PR curves, and the F1 result value of the reconciliation function for the overall evaluation.

The results of the three algorithms in terms of accuracy as the number of iterations increases are shown in Figure 7.

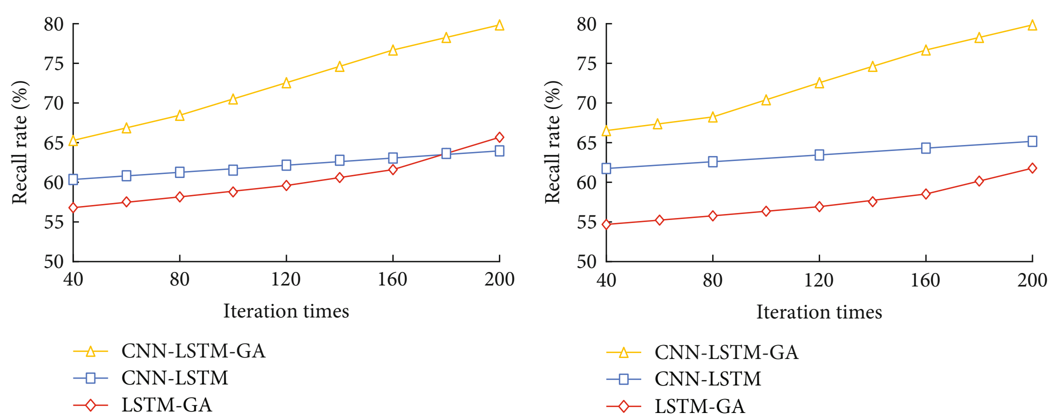
Table 4: Performance comparison results of the four algorithms.

Scale	Item	Reference [15]	Reference [16]	Research algorithm
4 × 5	AFI	0.241	0.172	0.110
	ASI	0	-0.281	-0.341
	AFS	0.130	-0.056	-0.115
10 × 10	AFI	0.181	0.153	0.112
	ASI	-0.092	-0.24	-0.784
	AFS	0.090	-0.047	-0.332
10 × 15	AFI	0.121	0.074	-0.055
	ASI	-0.084	-0.351	-0.623
	AFS	0.025	-0.147	-0.285



(a) Testing set (b) Verify set

Figure 7: Accuracy rate results of three algorithms along with times.

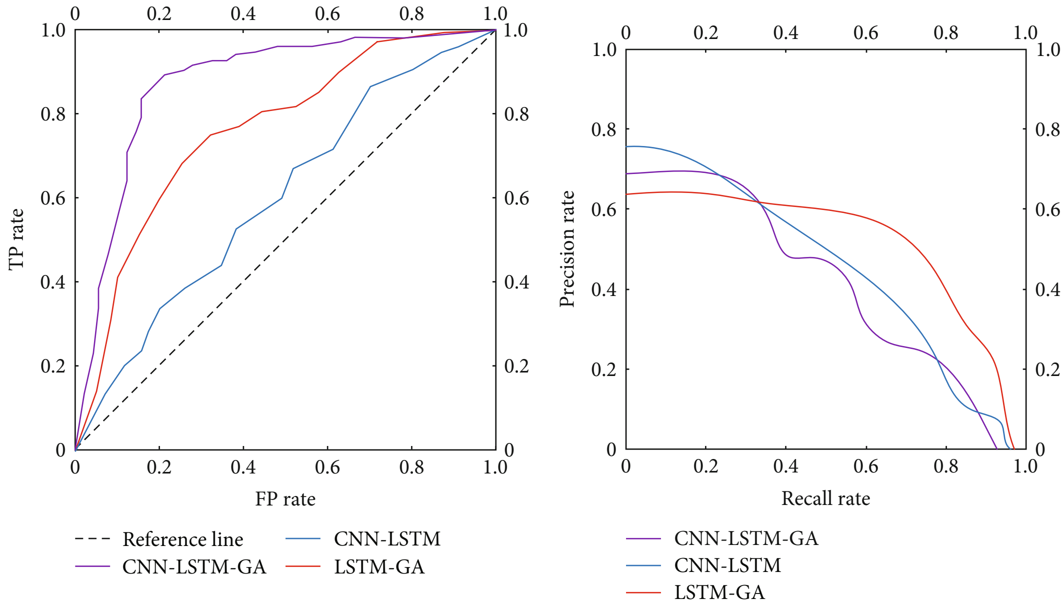


(a) Testing set (b) Verify set

Figure 8: Recall rate results of three algorithms along with times.

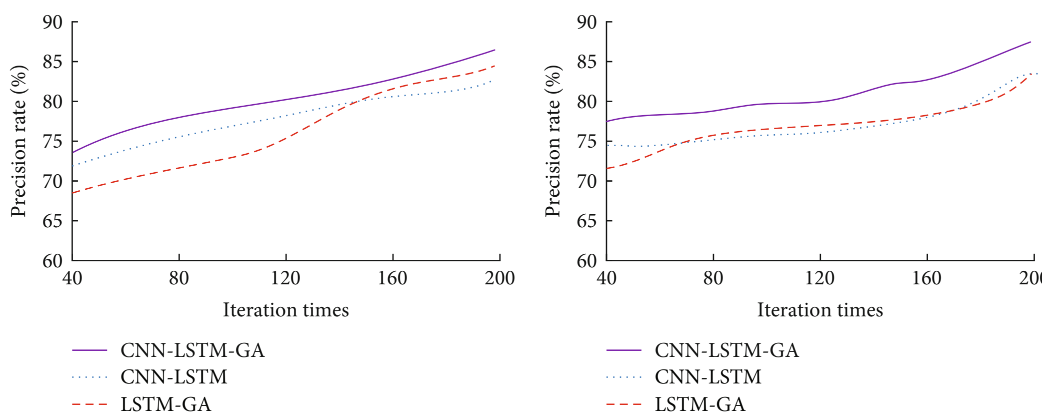
In Figure 7, in the test set and verification set, the accuracy of each algorithm presents an upward trend with the increase of the number of iterations. And the accuracy of CNN-LSTM-GA algorithm is significantly higher than the other two algorithms. Figure 6(a) shows that the accuracy of CNN-LSTM-GA algorithm ranges from 85.2% to 95.3%, while the accuracy of CNN-LSTM and LSTM-GA algorithms ranges from 67.5% to 79.9%. The lowest accuracy of the CNN-LSTM-GA method in the verification set is 84.6%, as shown in Figure 6(a), while the highest accuracy of the other two algorithms is roughly 71.5%. The enhanced CNN-LSTM-GA shows clear performance gains in terms of the accuracy of positive case verification, to sum up.

In the simulation experiment results, the variation of the recall rate of the three algorithms with the number of iterations is shown in Figure 8.



(a) ROC curve (b) PR curve

Figure 9: ROC curve and PR curve of four algorithms.



(a) Testing set (b) Verify set

Figure 10: Precision rate results of three algorithms along with times.

In Figure 8, the recall rate of CNN-LSTM-GA is significantly higher than that of the other two algorithms in both the test and verification sets. In Figure 7(a), the minimum recall rate of the CNN-LSTM-GA algorithm is 65.08%. Its accuracy tends to increase with the number of iterations and reaches the maximum at the number of iterations of 200, which is 79.8%. The maximum recall rates of CNN-LSTM algorithm and LSTM-GA algorithm are 63.6% and 64.8%, respectively. The maximum recall rate of these two algorithms in the test set is smaller than the minimum recall rate of CNN-LSTM-GA algorithm. In Figure 7(b), the recall rate of CNN-LSTM-GA algorithm ranges from 66.8% to 79.8%. The maximum recall rates of CNN-LSTM algorithm and LSTM-GA algorithm are 62.7% and 60.9%, respectively. In summary, the results showed that CNN-LSTM-GA had obvious performance advantages over more traditional algorithms in the numerical judgment of negative cases.

The results of the ROC curves as well as the PR curves for the combined test and validation sets of the three algorithms are shown in Figure 9.

Figure 9(a) shows that in the ROC curve results, the CNN-LSTM-GA algorithm has a much higher area under the curve than the other two algorithms. And the AUC value of CNN-LSTM-GA algorithm is 0.92. The AUC values of CNN-LSTM algorithm and LSTM-GA algorithm are 0.74 and 0.65, respectively. It shows that the prediction performance of the research algorithm is better. As can be seen in Figure 9(b), in the PR curve results, the area under the curve of CNN-LSTM-GA is smaller than that of the other two algorithms. It indicates that the prediction deviation of NN-LSTM-GA algorithm is smaller than that of the other two algorithms. In conclusion, the ratio of positive to negative cases is disproportionately high. The combined ROC and PR curve findings demonstrate that CNN-LSTM-GA

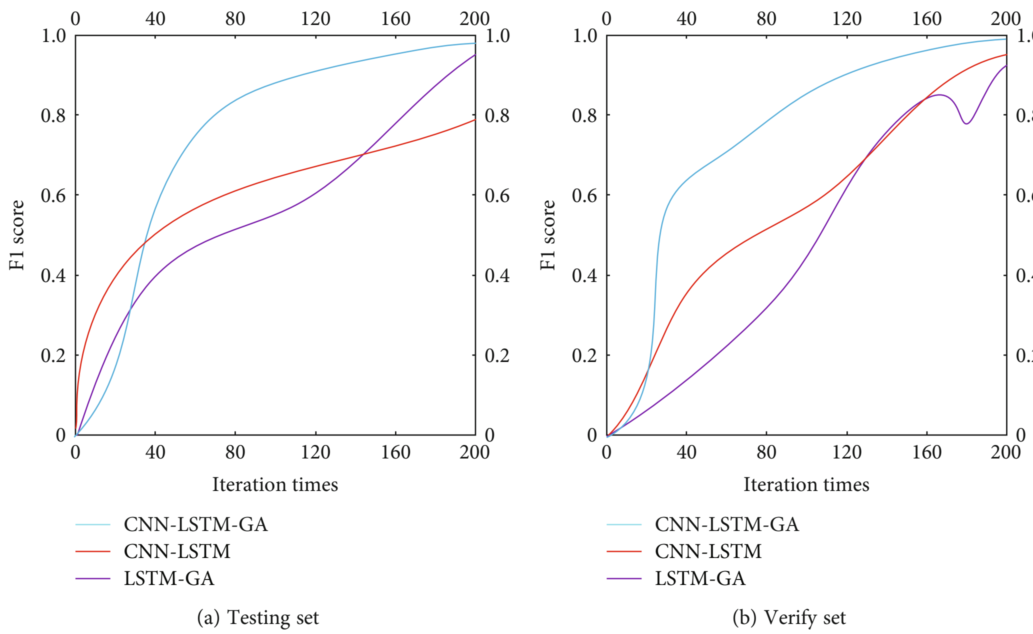


FIGURE 11: $F1$ score results of three algorithms along with times.

has a clear overall performance advantage in data prediction and solution.

The accuracy rates of each algorithm are shown in Figure 10 as the number of iterations rises.

In Figure 10, the curve for CNN-LSTM-GA is always on top of the three algorithms, indicating that this algorithm is consistently higher than the other two algorithms in terms of accuracy rate results. The average results obtained were 79.98%, 71.24%, and 69.03% for the three algorithms CNN-LSTM-GA, CNN-LSTM, and LSTM-GA, respectively. The average value of CNN-LSTM-GA was found to be significant when compared to the other three average values using significance analysis. It shows that the algorithm performs much better in terms of performance and the fineness of the overall validation of positive and negative cases than the other two algorithms.

The results of the three algorithms are combined, and the value of the harmonic function F1 changes with the number of iterations. The simulation results of solving the flexible job-shop scheduling problem are shown in Figure 11.

In Figure 11(a), after the number of iterations reaches 40, CNN-LSTM-GA algorithm maintains its $F1$ score above 0.8, which is significantly higher than the other two algorithms. Figure 11(b) shows that, for the same number of iterations, the CNN-LSTM-GA method's $F1$ score is higher than those of the CNN-LSTM algorithm and the LSTM-GA algorithm. The $F1$ score of the CNN-LSTM-GA method also approaches 0.98 when the number of iterations exceeds 200. In conclusion, it demonstrates that in terms of both overall performance and balance performance, CNN-LSTM-GA clearly outperforms the other two algorithms. To study the performance of the algorithm by horizontal comparison, the FJSP methods of literatures [32–35] were tested in the test set. The values of the test results are compared. As shown in Table 5, these are the results of sensitivity (Ae), specificity (Se), accuracy (Ac), and mean square error (MSE) of each algorithm.

TABLE 5: Comparison of test results of each method.

Algorithm	Ae	Se	Ac	MSE
BP [32]	0.82	0.81	0.82	0.01118
CNN [33]	0.86	0.86	0.84	0.00097
PSO-CNN [34]	0.91	0.91	0.88	0.00089
GA-CNN [35]	0.93	0.92	0.91	0.00132
CNN-LSTM-GA	0.95	0.96	0.93	0.000820

As shown in Table 1, the sensitivity, specificity, and accuracy of the CNN-LSTM-GA algorithm studied were 0.95, 0.96, 0.93, and 0.000820, respectively. Among all the algorithms, the mean square error of the research method is the smallest. Meanwhile, its sensitivity, specificity, and accuracy are the highest among them. It can be seen that compared with the current advanced FJSP model, the FJSP model based on research algorithm has the best performance.

5. Discussion

In this study, FJSP is elaborated in detail, in order to make FJSP have good robustness and minimize the impact of machine failure. This paper presents an improved genetic algorithm to solve FJSP and mainly presents a prediction model constructed by CNN and LSTM. And the corresponding scheduling flow model is designed. The results showed that the proposed method can evaluate the robustness of scheduling more efficiently and accurately and is effective for solving FJSP. In conclusion, this paper chooses a successful scheduling method and fully exploits the workshop state data to create a precise prediction model of workshop events. The global manufacturing industry is moving into the era of information technology and intelligence. Under this trend, intelligent manufacturing is becoming more and more influential in the manufacturing industry. There are still a lot of issues in the future work that are worth researching. First off, the FJSP method's workshop dynamic event response mode is very straightforward, but the level of predictive processing is insufficient. The aforementioned strategy can still be used when there are not many dynamic events. However, it is challenging to cope with the projected dynamic occurrences better when dynamic events happen frequently. The research of the dynamic event prediction method and rescheduling strategy needs to be further investigated in order to decrease the postadjustment frequency of the scheduling plan and better adapt to the dynamic job-shop environment.

6. Conclusion

In the actual scenarios, it is necessary to make sure that the flexible workshop production can proceed smoothly and swiftly, as well as to resolve the multiobjective FJSP problem. Therefore, LSTM, CNN, and GA algorithms are combined in this study to create a novel model and an enhanced CNN-LSTM-GA algorithm. This algorithm is integrated with CNN-LSTM and LSTM-GA, and the same data is used to obtain comparative test results. In the test set, the accuracy of the CNN-LSTM-GA algorithm was between 85.2% and 95.3%, which was significantly higher than the other two algorithms. In the verification set, the minimum accuracy of the CNN-LSTM-GA algorithm was 84.6%, both higher than the maximum accuracy of the other two algorithms. The recall rate of CNN-LSTM-GA was also significantly higher than that of the other two algorithms. The results showed that CNN-LSTM-GA had clear performance advantages over the traditional algorithms in numerically judging negative cases. In the FJSP simulation experiment, the AUC value of the CNN-LSTM-GA algorithm was 0.92. After the number of iterations reached 40, the $F1$ value of the CNN-LSTM-GA algorithm remained above 0.8, which was significantly higher than the other two algorithms. When the number of iterations reached 200, the $F1$ score of CNN-LSTM-GA algorithm was close to 0.98. CNN-LSTM-GA was superior to the other two algorithms in the prediction accuracy and overall performance of FJSP. Genetic algorithms optimized based on deep learning neural networks can make better choices according to the current environment. And through continuous learning, it can adapt to the dynamic event interference in the new environment and has strong robustness. The research approach significantly outperforms the conventional genetic algorithm in terms of convergence, diversity, and stability and may be utilised to solve FJSP problems quickly and effectively. Although the research has made some achievements, the model and comparison algorithm constructed in this research are mainly aimed at the monitoring problem of FJSP. Additionally, the test results are rather straightforward, which is the primary issue that has to be resolved in future research. Additionally, various discrete combinatorial optimisation issues are amenable to the suggested algorithm's use. Chaotic crossover operators and chaotic mutation operators will be incorporated into the algorithm in the following stage. In order to improve computational outcomes, the hybrid algorithm's evolution process will also incorporate the notion of cultural evolution.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declared that this article is free of conflict of interest.

References

[1] M. Wu, D. Yang, and T. Liu, "An improved particle swarm algorithm with the elite retain strategy for solving flexible job-shop scheduling problem," *Journal of Physics: Conference Series*, vol. 2173, no. 1, pp. 12082–12099, 2022.

[2] K. Lei, P. Guo, W. Zhao et al., "A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem," *Expert Systems with Applications*, vol. 205, article 117796, 2022.

[3] Z. Chen, L. Zhang, X. Wang, and P. Gu, "Optimal Design of Flexible Job Shop Scheduling Under Resource Preemption Based on Deep Reinforcement Learning," *Complex System Modeling and Simulation*, vol. 2, no. 2, pp. 174–185, 2022.

[4] C. Wang, Y. Li, and X. Li, "Solving flexible job shop scheduling problem by a multi-swarm collaborative genetic algorithm," *Systems Engineering and Electronics: English Version*, vol. 32, no. 2, pp. 261–271, 2021.

[5] A. Shafiq, A. B. Olak, and T. N. Sindhu, "Designing artificial neural network of nanoparticle diameter and solid–fluid interfacial layer on single-walled carbon nanotubes/ethylene glycol nanofluid flow on thin slendering needles," *International Journal for Numerical Methods in Fluids*, vol. 93, no. 12, pp. 3384–3404, 2021.

[6] A. Shafiq, A. B. Olak, C. Swarup, and A. Showkat, "Reliability analysis based on mixture of Lindley distributions with artificial neural network," *Advanced Theory and Simulations*, vol. 5, no. 8, pp. 1–16, 2022.

[7] U. Shahzad, "A comparative analysis of artificial neural network and support vector machine for online transient stability prediction considering uncertainties," *Australian Journal of Electrical and Electronics Engineering*, vol. 19, no. 2, pp. 101–116, 2022.

[8] A. S. Fard, "Comparative study on performance of different artificial neural network methods for prediction of the COVID19," *Foresight: The Journal for Future Studies, Strategic Thinking and Policy*, vol. 24, no. 3/4, pp. 545–561, 2022.

[9] D. Szarek, "Neural network-based anomalous diffusion parameter estimation approaches for Gaussian processes," *International Journal of Advances in Engineering Sciences and Applied Mathematics*, vol. 13, no. 2-3, pp. 257–269, 2021.

[10] N. Chea and Y. Nam, "Classification of fundus images based on deep learning for detecting eye diseases," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 411–426, 2021.

[11] Y. Zheng, X. Li, Y. Si, W. Qin, and H. Tian, "Hybrid deep convolutional neural network with one-versus-one approach for solar flare prediction," *Monthly Notices of the Royal Astronomical Society*, vol. 507, no. 3, pp. 3519–3539, 2021.

[12] J. Bhola, S. Soni, and G. K. Cheema, "Genetic algorithm based optimized leach protocol for energy efficient wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1281–1288, 2021.

[13] Z. Xue, "Routing optimization of sensor nodes in the Internet of things based on genetic algorithm," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25142–25150, 2021.

[14] S. K. Sahu, "Novel ensemble model with genetic algorithm and principal components analysis for classification of chronic kidney disease," *International Journal of Applied Evolutionary Computation*, vol. 12, no. 4, pp. 1–17, 2021.

[15] G. Gong, Q. Deng, X. Gong, and D. Huang, "A non-dominated ensemble fitness ranking algorithm for multi-objective flexible job-shop scheduling problem considering worker flexibility and green factors," *Knowledge-Based Systems*, vol. 231, no. 14, pp. 107430–107449, 2021.

[16] W. Tan, X. Yuan, J. Wang, and X. Zhang, "A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced NSGA-II: an application from casting workshop," *Computers and Industrial Engineering*, vol. 160, no. 10, pp. 107557–107578, 2021.

[17] Y. Wang, Y. C. Song, Y. J. Zou, Q. Lei, and X. K. Wang, "A hybrid gray wolf weed algorithm for flexible job-shop scheduling problem," *Journal of Physics: Conference Series*, vol. 1828, no. 1, pp. 12162–12172, 2021.

[18] J. Guo, D. Lei, and M. Li, "Two-phase imperialist competitive algorithm for energy-efficient flexible job shop scheduling," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 6, pp. 12125–12137, 2021.

[19] K. Tamssaouet, P. S. Dauzère, S. Knopp, B. Abdoul, and Y. Claude, "Multiobjective optimization for complex flexible job-shop scheduling problems," *European Journal of Operational Research*, vol. 296, no. 1, pp. 87–100, 2022.

[20] X. Shen, J. Lu, X. You, L. Son, and Z. Ge, "A region enhanced discrete multi-objective fireworks algorithm for low-carbon vehicle routing problem," *Modeling and Simulation of Complex Systems*, vol. 2, no. 2, pp. 142–155, 2022.

[21] A. Javadi, M. Ghahremanzadeh, M. Sassi, O. Javanbakht, and B. Hayati, "Economic evaluation of the climate changes on food security in Iran: application of CGE model," *Theoretical and Applied Climatology*, vol. 151, no. 1-2, pp. 567–585, 2023.

[22] A. A. E. E. Adel, E. S. Ragab, M. S. Abdullah, and S. S. Ayman, "Economic and reliable preventive maintenance scheduling in power systems by using binary crow search algorithm," *International journal of Engineering Research in Africa*, vol. 56, pp. 182–198, 2021.

[23] F. Dharma, S. Shabrina, A. Noviana, M. Tahir, N. Hendrastuty, and W. Wahyono, "Prediction of Indonesian inflation rate using regression model based on genetic algorithms," *Jurnal Online Informatika*, vol. 5, no. 1, pp. 201–215, 2020.

[24] D. Carmen, A. Saldiva, and S. N. Elian, "A comparison of the estimators of the scale parameter of the errors distribution in the L₁ Regression," *Journal of Statistics*, vol. 12, no. 2, pp. 261–276, 2022.

[25] S. W. Wang, H. Q. Cao, and Y. Liu, "Application of SIFT algorithm based on the Gabor features in multi-source information image monitoring," *IEIESPC (IEIE Transactions on Smart Processing and Computing)*, vol. 12, no. 2, pp. 112–121, 2023.

[26] A. Ghorbel and B. Souissi, "Upper confidence bound integrated genetic algorithm-optimized long short-term memory network for click-through rate prediction," *Applied Stochastic Models in Business and Industry*, vol. 38, no. 3, pp. 475–496, 2022.

[27] R. Pahi, Z. Lonarevi, A. Gams, and A. Ude, "Robot skill learning in latent space of a deep autoencoder neural network," *Robotics and Autonomous Systems*, vol. 135, no. 14, pp. 103690–103700, 2021.

[28] O. A. Fadipe-Joseph, M. O. Oluwayemi, and E. O. Titiloye, "Subclasses of univalent functions involving modified sigmoid function," *International Journal of Difference Equations*, vol. 16, no. 1, pp. 465–506, 2021.

[29] L. Tach, M. B. Morrissey, E. Day, and B. Mihalec-Adkins, "Experiences of trauma-informed care in a family drug treatment court," *Social Service Review*, vol. 96, no. 3, pp. 465–506, 2022.

[30] H. Deng, W. X. Zhang, and Z. F. Liang, "Application of BP neural network and convolutional neural network (CNN) in bearing fault diagnosis," *IOP Conference Series: Materials Science and Engineering*, vol. 1043, no. 4, article 042026, 2021.

[31] Y. X. Yang, P. Wang, X. J. Gao, H. H. Gao, and Z. Y. Qi, "Research and application of RBF neural network based on modified Levenberg-Marquardt," *Journal of Computational Methods in Sciences and Engineering*, vol. 22, no. 5, pp. 1597–1619, 2022.

[32] A. Stanković, G. Petrović, D. Marković, and Ž. Ćojbašić, "Solving flexible job shop scheduling problem with transportation time based on neuro -fuzzy suggested metaheuristics," *Acta Polytechnica Hungarica*, vol. 19, no. 4, pp. 209–227, 2022.

[33] D. Tremblet, S. Thevenin, and A. Dolgui, "Predicting makespan in flexible job shop scheduling problem using machine learning," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 1–6, 2022.

[34] L. M. Thi, A. T. T. Mai, and H. N. Van, "An improved hybrid metaheuristics and rule-based approach for flexible job-shop scheduling subject to machine breakdowns," *Engineering Optimization*, vol. 55, no. 9, pp. 1535–1555, 2023.

[35] G. G. Wang, D. Gao, and W. Pedrycz, "Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8519–8528, 2022.